MAA*: Un algorithme de recherche heuristique pour la résolution exacte de DEC-POMDPs

Daniel Szer, François Charpillet, Shlomo Zilberstein

MAA* : Un algorithme de recherche heuristique pour la résolution exacte de DEC-POMDPs

Daniel Szer et **François Charpillet**
MAIA Group
INRIA Lorraine, LORIA
54506 Vandœuvre-lès-Nancy, France
{szer,charp}@loria.fr

Shlomo Zilberstein
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
shlomo@cs.umass.edu

Résumé

Nous présentons ici MAA*, le premier algorithme de recherche heuristique à la fois complet et optimal pour résoudre des processus de décision markovien décentralisés (DEC-POMDPs) à horizon fini. Il permet de calculer des plans optimaux pour un groupe d'agents coopératifs dans un environnement stochastique et partiellement observable. La résolution de tels problèmes est particulièrement dur, mais permet d'aborder des domaines importants tels que le contrôle de robots autonomes. Notre approche consiste en une synthèse entre des méthodes de recherche heuristique et la théorie du contrôle décentralisé, et nous sommes capables de montrer qu'elle présente des avantages intéressants vis-à-vis des solutions existantes.

1 Introduction

Nous nous intéressons dans ce travail au problème de planification optimale pour un groupe d'agents coopératifs dans un environnement incertain et partiellement observable. On peut rencontrer de tels problèmes dans des domaines comme le routage de paquets dans les reseaux de communication, les chaînes d'approvisionnements distribuées, le contrôle de robots sauveteurs ou celui de robots explorateurs dans l'espace.

Un cadre formel pour décrire des problèmes de décision distribués a été établi récemment avec le modèle DEC-POMDP [1]. Un DEC-POMDP à n-agents est donné par $\langle S, \{A_i\}, P, R, \{\Omega_i\}, O, T, s_0\rangle$, où S est un ensemble fini d'états, A_i est un ensemble fini d'actions pour l'agent i, $P(s, a_1, \dots a_n)$ est une fonction de probabilités de transition, $R(s, a_1, \dots a_n)$ est une fonction de récompense, Ω_i est un ensemble fini d'observations pour agent i, $O(s, a_1, \dots a_n, o_1, \dots o_n)$ est une fonction de probabilités d'observation, T est l'horizon du problème et s_0 est l'état initial du système. Résoudre un DEC-POMDP signifie en général trouver un ensemble de *politiques*, une pour chaque agent, tel que leur exécution parallèle maximize $E[\sum_{t=1}^{T} R(s_t, \vec{a}_t)|s_0]$, l'espérance de la somme des récompenses associées à chaque transition.

La complexité du DEC-POMDP est NEXP-complet [1], contrairement au cas POMDP mono-agent, qui lui est PSPACE-dur. Une première solution optimale, étant basée sur la programmation dynamique, a été proposée récemment dans [2] pour les jeux stochastiques partiellement observables.

2 L'algorithme MAA*

Une politique optimale pour un POMDP à horizon fini t peut être représentée comme un arbre de décision q^t de profondeur t, où les noeuds représentent les actions à effectuer, et où le branchement est choisi en fonction des observation reçues. De façon similaire, une solution pour un DEC-POMDP peut être représentée par un *vecteur de politiques* $\delta^t = (q_1^t, \dots q_n^t)$, où l'abre q_i^t sera executé par l'agent i. Nous posons $V(s_0, \delta)$ comme la somme des récompenses espérées quand le vecteur de politiques δ est executé à partir de l'état s_0. Elle peut être calculée facilement à partir des paramètres du problème.

Notre approche est l'extension multi-agent de l'algorithme de recherche heuristique A* : Nous sommes en effet capables de construire progressivement un arbre de recherche dans l'espace des vecteurs de politiques, en supprimant les parties de l'arbre qui sont dominées par d'autres. Un noeud à profondeur t dans l'abre de recherche contient un vecteur de politiques à horizon t, et chaque itération de l'algorithme consiste à évaluer la liste D des feuilles de l'arbre, à choisir la feuille la plus prometteuse, et de la développer un niveau de plus.

La partie essentielle de A* est la décomposition de la fonction d'évaluation en une partie exacte d'un début de solution et une estimation optimiste (= l'heuristique) pour la partie restante. Dans notre cas, il s'agit donc d'évaluer des vecteurs de politique δ^t pour un certain horizon t, et de trouver une heuristique H^{T-t} pour évaluer ce qui se passe après l'exécution de δ^t :

$$F^T(s_0, \delta^t) = V(s_0, \delta^t) + H^{T-t}(s_0, \delta^t) \quad (1)$$

On dit qu'une heuristique est *admissible* si elle surestime la valeur de toute politique réelle. Surestimer la valeur d'un DEC-POMDP peut se faire de multiples façons, en utilisant par exemple les solutions du POMDP ou même du MDP sous-jacent, qui eux sont plus faciles à calculer. Pour toute heuristique admissible, nous avons le théorème suivant :

Théorème 2.1. *MAA* est complet et optimal.*

Initialiser la liste $D_0 = \times_i A_i$
Pour chaque itération i :

1. Choisir $\delta^* \in D_i$ avec $\delta^* \neq \delta^T$ tel que $\forall \delta \in D_i : F^T(s_0, \delta) \leq F^T(s_0, \delta^*)$
2. $\delta^{*'} \leftarrow$ Développer δ^*
3. Si $\delta^{*'}$ est une solution améliorée :
   (a) Afficher $\delta^{*'}$
   (b) $\forall \delta \in D_i$: Si $F^T(s_0, \delta) \leq F^T(s_0, \delta^{*'}), D_i \leftarrow D_i \setminus \delta$
4. $D_i \leftarrow D_i \cup \delta^{*'}$
5. Si δ^* est complètement développé, $D_i \leftarrow D_i \setminus \delta^*$

jusqu'à ce que $\exists \delta^T \in D_i$ avec $\forall \delta \in D_i : F^T(s_0, \delta) \leq F^T(s_0, \delta^T) = V(s_0, \delta^T)$

FIG. 1 – L'algorithme MAA*

3 Résultats, conclusions et perspectives

Des expériences préliminaires ont montré que MAA* est souvent plus performant aussi bien en temps de calcul qu'en utilisation de mémoire que les solutions existantes, en l'occurrence l'algorithme de programmation dynamique de [2]. La performance dépend surtout de l'efficacité de l'heuristique : Plus celle-ci est proche de la vraie fonction de valeur, plus on est capable d'exclure des parties de l'arbre de recherche.

Nous sommes en train de généraliser notre approche pour pouvoir traîter les DEC-POMDPs à horizon infini. Dans ce cas, il ne s'agira plus de calculer des solutions exactes, et la recherche se fera alors dans l'espace des automates finis.

Références

[1] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27(4) :819–840, 2002.

[2] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th AAAI*, 2004.